

# Computing minimum area homologies

Erin Wolf Chambers\*

Mikael Vejdemo-Johansson†

## Abstract

Calculating and categorizing the similarity of curves is a fundamental problem which has generated much recent interest. However, to date there are no implementations of these algorithms for curves on surfaces with provable guarantees on the quality of the measure. In this paper, we present a similarity measure for any two cycles that are homologous, where we calculate the minimum area of any homology (or connected bounding chain) between the two cycles. The minimum area homology exists for broader classes of cycles than previous measures which are based on homotopy. It is also much easier to compute than previously defined measures, yielding an efficient implementation that is based on linear algebra tools. We demonstrate our algorithm on a range of inputs, showing examples which highlight the feasibility of this similarity measure.

## 1 Introduction

Recently, much work has been done in the area of similarity measures for curves in topological spaces. For example, in the graphics, vision, and medical imaging communities, the goal is often to measure similarity between objects [SHI96, TV04], with the goal of categorizing or recognizing similar structures. Questions regarding similarity between maps or trajectories within a map are motivated by the vast quantity of data from GIS systems; see for example [AKPW14] for a recent survey of some algorithms in this area.

In the computational geometry community, much of the focus has been on the Fréchet distance in Euclidean space [AG95, Alt09] and its generalizations such as geodesic and homotopic Fréchet distance [CW10, CCE\*10]. However, none of these have proven algorithmically tractable to compute on surfaces. There is also work giving an algorithm to compute the minimum area homotopy between two curves or cycles on surfaces; somewhat surprisingly, computing area is much more practical, giving algorithms which are either quadratic or linear in time (depending on the exact setting) [CW13]. Although this minimum homotopy algorithm has not been implemented, it is the first theoretically tractable algorithm in the surface setting.

The use of homology rather than homotopy was suggested in prior work as a natural measure to consider [CW13]; this is perhaps especially appealing given that homology computations often have extremely efficient implementations which are based on operations over vector spaces. For example, there are efficient algorithms which compute the minimum cycle homologous to an input cycle [DHK11]. In addition, there is a large body of work on calculating homology groups and generators in a variety of settings. In addition to being a standard exercise in linear algebra, tools for computing simplicial homology – either of triangulations or induced from point cloud data – are commonplace, and occur in software packages including GAP [GAP14, EG08, VJ12], Magma [BCP97], and Sage [SJ05], as well as more specialized libraries such as ChomP [KMM04], javaPlex [TVJA12], Dionysus [Mor12], Perseus [Nan12], pHat [BKR13] and Kenzo [RS02].

However, the concept of finding a minimum area homology has not yet been investigated from the algorithmic perspective. This is perhaps due to the fact that homotopy describes a more natural notion of a continuous deformation, while with homology it is much less clear what geometric structure is described. There is a clear connection between the two; indeed, for higher dimensional manifolds, the problem of finding a minimum homotopy reduces to that of finding a minimum homology [Whi84]. Moreover, for the case of cycles on a surface, two homotopic cycles clearly

\*Dept. of Math and Computer Science, Saint Louis University, [echambe5@slu.edu](mailto:echambe5@slu.edu). Research supported in part by the NSF under Grant No. CCF-1054779 and IIS-1319573.

†Institute for Mathematics and its Applications, University of Minnesota, Minneapolis. Computer Vision and Active Perception Laboratory, KTH Royal Institute of Technology, Stockholm. AI Laboratory, Jozef Stefan Institute, Ljubljana. [mvj@kth.se](mailto:mvj@kth.se). Research supported in part by the FP7 project TOPOSYS (FP7-ICT-318493-STREP).

have a homology between them, and actions such as puncturing the surface or introducing high weight simplices will often force the minimum area homology to agree with the minimum homotopy.

In this paper, we formalize a notion of a homology between two cycles with minimum area, and prove several interesting properties. We also implement algorithms to compute this area, concluding with a series of examples to demonstrate the success of this measure.

It is also worth noting that by working with homology instead of homotopy, we can construct bounding chains for a wider class of structures on a surface. Recall that two (not necessarily connected) dimension  $k$  submanifolds  $S, T$  of a manifold  $M$  are said to be *cobordant* if there is a dimension  $k + 1$  submanifold  $W$  of  $M$  with boundary  $S \sqcup T$ . The connecting submanifold  $W$  is said to be the *cobordism* from  $S$  to  $T$ . Cobordisms are one of the most important techniques in contemporary algebraic and differential topology, and were fundamentally important in some of the most famous theorems in the field, including the Hirzebruch-Riemann-Roch theorem [HSB66] and the Atiyah-Singer index theorem [AS71]. A fundamental survey of cobordisms and their uses was written by Milnor in 1962 [Mil07].

For two arbitrary closed collections  $S, T$  of disjoint simple curves on a surface, computing a connecting chain generates a cobordism from  $S$  to  $T$ . So while we focus solely on minimum area homologies in this work, our algorithm to compute a connecting chain generalizes to significantly more complex structures in higher dimensions.

## 2 Definitions

In this paper we make extensive use of simplicial homology for triangulated surfaces, and its properties. We provide a brief overview of this area, and refer to the reader to a standard reference on topology for full details [Hat02, Mun00].

Recall that given a triangulation of a surface  $M$  with vertices  $V$ , edges  $E$  and faces  $F$ , there are *chain vector spaces*  $\mathbb{R}^V, \mathbb{R}^E, \mathbb{R}^F$ . Vectors in these vector spaces correspond to weighted collections of faces, edges, vertices. There is a linear *boundary operator*  $\partial_2 : \mathbb{R}^F \rightarrow \mathbb{R}^E$  and  $\partial_1 : \mathbb{R}^E \rightarrow \mathbb{R}^V$ . Closed curves composed out of the triangulation edges are represented by vectors in  $\ker \partial_1$ , and curves that occur as boundaries of collections of triangles are closed curves and are in  $\text{img } \partial_2$ , so that  $\text{img } \partial_2 \subseteq \ker \partial_1$ . The vector space of *essential closed curves* is represented by the homology group  $H_1(M; \mathbb{R}) = \ker \partial_1 / \text{img } \partial_2$ . In particular, we call two vectors of edge combinations  $z, w$  *homologous* if  $z - w \in \text{img } \partial_2$ ; in this case,  $z$  and  $w$  represent the same essential closed curve.

We can now consider a *minimum area homology* between  $z$  and  $w$  - namely, a connected bounding chain of triangles  $c$  where  $\partial c = z - w$  and the area of  $c$  is the infimum over all possible such bounding chains. Here, and later in the paper, we use the term *bounding chain* to denote precisely some  $c$  such that  $\partial c = z - w$ . For instance, a simple closed curve on the surface not encycling any tunnels or holes would have the interior disk as its bounding chain. This minimum area measure between curves is naturally commutative and satisfies a notion of the triangle inequality, since homology is an equivalence relation, and therefore transitive.

Homotopy, the more usual measure of similarity in this area, is perhaps more intuitive, since a *homotopy* between two cycles is simply a continuous deformation between them. Homology is a coarser measure than homotopy, since the homology group is a abelization of the homotopy group; any homotopic cycles are necessarily homologous, but the reverse is not true. However, when calculating area, it is not clear that minimum area homotopies even exist, and great care is taken in the mathematical literature to restrict the type of integral in order to guarantee existence; see the book by Lawson for an overview of this problem and its historical developments [Law80].

In contrast, minimum area homologies automatically exist (so the infimum becomes a minimum) and are computable given the fact that we are working in a vector space. In addition, homology has the added advantage that we can consider more than simple cycles which are continuously deformable to each other, since homology allows one to deform “over” handles and other topological features. See Figure 1 for an example of homologous cycles on a surface which are not homotopic, as well as Figures 6 and 7.

## 3 Connection to homotopy

In this section, we first focus on the case of finding the minimum area homology for two homotopic input cycles, since we expect this to be the most interesting application of our algorithm. The following lemmas establish existence of minimum area homologies in this case and the connection between the minimum area homology and minimum area homotopy.

**Lemma 1.** *Suppose we have two cycles  $z, w : S^1 \rightarrow M$  on a manifold where  $z$  is homotopic to  $w$ . Then there exists a connected bounding chain  $c$  such that  $\partial c = z - w$ .*

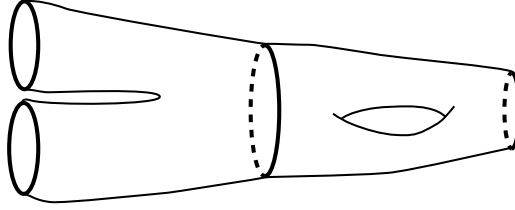


Figure 1: Two examples of homologous cycles on a surface that are not homotopic. The middle cycle is homologous both to the right cycle and to the sum of the left two cycles; the rightwards homology is not a homotopy due to the hole in the surface, the leftwards one due to the split of one cycle into two. We will see examples of both behaviours later in the paper, in Figure 6.

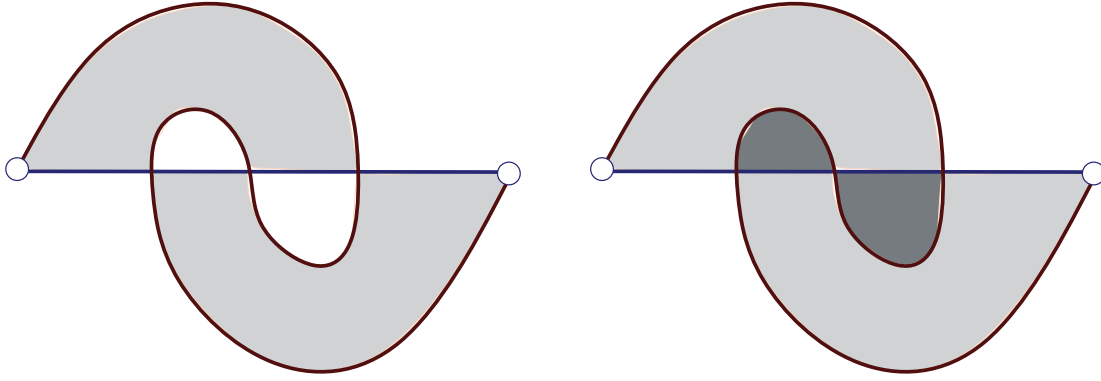


Figure 2: Top: The minimum area homology between these two curves consists of the sum of the shaded areas. Bottom: The minimum area homotopy, in contrast, must sweep regions "twice"; since homotopy does not cancel, the area of the small darkly shaded regions is counted twice in the area of the minimum homotopy, giving a strictly greater value. We revisit this type of example on a scanned mesh in Figure 7.

*Proof.* Since  $z$  and  $w$  are homotopic, there is a homotopy  $H : S^1 \times I \rightarrow M$  where  $H(\cdot, 0) = z$  and  $H(\cdot, 1) = w$ . We know that  $S^1 \times I$  is connected, so let  $c = \text{img}(H)$ .  $\square$

Note that if  $M$  is triangulated, both cycles consist of edges in the triangulation, and there is a homotopy built from the triangulation of  $M$ , then this lemma holds as is for the simplicial case.

The homology equivalence class of connected bounding chains in Lemma 1 is a finite dimensional affine linear space if  $M$  has a finite triangulation. The area measure is a continuous function to the non-negative reals, and thus attains its minimum. Therefore, we can refer to the minimum area homology as the bounding chain which attains the infimum in this area measure.

For well-behaved cases in a 2-sphere, we can in fact narrow the candidates down more concretely.

**Proposition 1.** *Consider two disjoint cycles  $z, w : S^1 \rightarrow S^2$  on a sphere, with  $z$  homotopic to  $w$ . Then there are only 2 supports for a connected bounding chain, one of which describes the minimum area homotopy.*

*Proof.* Consider two cycles  $z$  and  $w$  that are disjoint. Fix  $z$ , and the Jordan curve theorem immediately implies that there are two disjoint sides of the  $S^2 - z$ . Since  $w$  is disjoint, it is on one side or the other, so the bounding chain is either the region of  $S^2 - z - w$  that is "between"  $z$  and  $w$ , or is the bounding chain which is the union of the disk bounded by  $z$  and the disk bounded by  $w$ .  $\square$

As with Lemma 1, the result here holds as is if the cycles respect an existing triangulation of the sphere.

In the case that cycles intersect, the situation is more complex, since there are more than 2 connected bounding chains for such inputs. See [CW13] for a more thorough discussion of this complexity in the context of homotopy; since each homotopy class also gives a valid homology between the curves, the same issue exists in our setting.

Since the image of a homotopy produces a connected bounding chain, we can also observe that the minimum area homotopy is bounded below in area by the minimum area homology. However, there are cases (even for homotopy between simple curves in the plane) where the minimum area homotopy is strictly greater than the minimum area homology; see Figure 2.

## 4 Algorithm

Our approach to an algorithm and an implementation for the minimum homology area is rooted in the extensive work available on finding small solutions to linear systems of equations.

### 4.1 Optimization problem formulation

Computing the minimum area bounding chain for a cycle  $z \in C_1 M$  (where here we use  $z$  to denote the difference of the two input cycles rather than one of the two inputs) is a question of computing a minimum area solution to a linear set of equations. A bounding chain for  $z$  is some  $x$  such that  $\partial x = z$ . Hence, we can formulate the minimum area bounding chain by solving the optimization problem

$$\arg \min_x \sum_j a(\sigma_j) |x_j| \quad \text{subject to } \partial x = z \quad (1)$$

where  $a(\sigma_j)$  is the area contribution from cell  $\sigma_j$ .

There are several variations of this optimization problem that turn out to be useful for implementation, speed or practicality.

If we write  $A$  for the diagonal matrix with entries  $a(\sigma_j)$ , we note that  $\sum_j a(\sigma_j) |x_j| = \|Ax\|_1$ . We can recast Problem 1 as a pure  $L_1$ -minimization problem. To do this, we note that if  $x' = Ax$  and all areas are non-zero, then  $\partial x = \partial A^{-1}x'$ . Thus,  $x$  minimizing  $\|Ax\|_1$  subject to  $\partial x = z$  can be computed by first computing  $x'$  minimizing  $\|x'\|_1$  subject to  $\partial A^{-1}x' = z$ . Thus, we can also solve for the minimum area bounding chain by solving the optimization problem

$$\arg \min_x \|x\|_1 \quad \text{subject to } \partial A^{-1}x = z \quad (2)$$

This equation captures a known optimization problem in machine learning and compressed sensing [Don06], which can be solved i.a. using the Orthogonal Matching Pursuit (OMP) algorithm [MZ93, RZE08].

It is worth noting for the specific case of surface meshes in  $\mathbb{R}^3$  that have no boundaries, the results by [DHK11] offer up a linear programming approach to the optimization problem. Running in  $O(n^3)$ , this is an attractive option for an exact solution. However, this does not cover all the cases we consider in this paper, and restricts the generalizability to higher dimensions. As pointed out in [DHK11], the presence of a Möbius strip is sufficient to break the total unimodularity assumption that the linear programming approach relies on. (We would like to thank one of our anonymous reviewers for alerting us to the linear programming approach.)

### 4.2 Trade-off between $L_1$ and $L_2$

While the computation of the  $L_1$ -minimization problem in Section 4.1 is a clear way to produce an area-minimal connecting chain, this class of minimization problems lacks closed form analytic solutions. In this section, we shall describe a related  $L_2$ -minimization problem, as well as bounds that relate it to the area-minimization problem.

It is an immediate consequence of the Hölder inequality and the Cauchy-Schwarz inequality that  $\|x\|_2 \leq \|x\|_1 \leq \sqrt{n}\|x\|_2$  for any  $x \in \mathbb{R}^n$ . It follows that we can bound the area norm for a finite triangulation with a weighted  $L_2$ -norm.

One way to get an approximate solution to the minimum area is to compute  $\arg \min_{\hat{x}} \hat{x}^T \hat{x}$  such that  $\partial \hat{x} = \hat{z}$ , where  $\hat{x} = A^{1/2}x$ . Then  $\hat{x}^T x = x^T (A^{1/2})^T A^{1/2} x = x^T A x = \sum_j a(\sigma_j) x_j^2$ . This lets us solve  $\arg \min_{\hat{x}} \hat{x}^T \hat{x}$  such that  $\partial \hat{x} = \hat{z}$ , and then the appropriate  $x$  is  $A^{1/2} \hat{x}$ .

Replacing the  $L_1$  problems with  $L_2$  problems in this way produces these two formulations of area minimization as  $L_2$  minimization problems:

$$\arg \min_x x^T A x \quad \text{subject to } \partial x = z \quad (3)$$

$$\arg \min_x \|x\|_2 \quad \text{subject to } \partial A^{-1/2} x = z \quad (4)$$

A large benefit of this reformulation is that the  $L_2$ -problem 4 is the *least squares* optimization problem and has an analytic solution. Writing  $W = \partial A^{-1/2}$ , the vector  $z = (W^T W)^{-1} W^T w$  is a solution of minimum norm to  $Wz = w$ . Using the bounds we mentioned above, this  $L_2$ -minimizer is an approximation to the actual area minimal solution, and has the benefit of being computable in matrix multiplication time (generally written as  $O(n^\omega)$ ) with a small constant for contemporary implementations, where  $n$  is the larger of the counts of triangles  $T$  and edges  $E$  in the triangulation. OMP, as a comparison, can be implemented with parametric complexity  $O(n^\omega k + (E+T)k + k^3)$  where  $k$  is  $\|z\|_1$ .

### 4.3 Bounded area variation

It is worth noting that our initial implementations, which neglected to include any weight calculations but instead relied only on the number of simplices in the homology, worked quite well on many of the initial test cases. This is perhaps not terribly surprising, given that our initial inputs were well formed with relatively even sized triangles, so that comparing the number of triangles included a connected bounding chain formed a reasonable approximation to area. Given the simplicity of the this code and speed-up of computation, we formalize this idea as follows.

**Theorem 1.** *Suppose the area of all triangles  $\sigma$  in a triangulation of a surface  $M$  are bounded by  $a \leq A(\sigma) \leq A$ .*

*If  $z$  minimizes  $\|z\|_1$  subject to  $\partial z = w$  and  $z'$  minimizes  $A(z')$  subject to  $\partial z' = w$ , so that  $z$  is  $L_1$ -minimal and  $z'$  is area minimal, then  $a\|z\|_1 \leq A(z') \leq A\|z\|_1$ .*

*Proof.* Since  $z$  is  $L_1$ -minimal, we know that  $\|z\|_1 \leq \|z'\|_1$ . Since  $a$  is a lower bound of the triangle areas, this establishes the lower bound.

Since  $z'$  is area minimal and  $A$  is an upper bound on triangle areas, it follows that  $A(z') \leq A(z) \leq A\|z\|_1$ . This establishes the upper bound.  $\square$

This means that even without solving for an area minimal bounding chain we are able to establish a confidence interval for the area minimal bounding chain; and unless the local triangulation density varies considerably, we are likely to find the area minimal bounding chain by simply minimizing the triangle count.

### 4.4 Higher dimensions

Our implementation in the next section focuses on triangulated meshes, simply because that is the data most readily available as well as the most immediate application. However, the algorithm we describe works for arbitrary dimensions of meshes and embedded substructures: curves or surfaces in volume meshes or higher dimensional analogues. For a  $k$ -dimensional submesh of a  $d$ -dimensional mesh, the area bounds in Theorem 1 as well as the entire analysis of the optimization problem formulation and its complexity carry over with the bounding chain taken to be a  $k+1$ -dimensional submesh. In this setting, “area” would be replaced by its  $k+1$ -dimensional correspondence.

## 5 Results

In our implementation, we store the areas (or weights) as a diagonal matrix  $A$ , and let  $D$  be the matrix which stores the boundary operator  $\partial$ . Our algorithm should compute  $\arg \min x^T A x$  such that  $Dx = z$ , where  $z$  is the difference of the two input cycles whose minimum area homology we wish to compute. In the following sections, we describe the various implementations and approximations we pursued, along with results from each.

### 5.1 Proof of concept implementation

We can modify our problem to look like a standard machine learning algorithm as follows. In orthogonal matching pursuit, the goal is to compute  $\arg \min \|\hat{x}\|_1$  such that  $D\hat{x} = \hat{z}$  [PRK93], and code to compute has been done for many languages, including the scikit-learn package in python [PVG\*11]. Here, we will let  $\hat{x} = Wx$ , so that

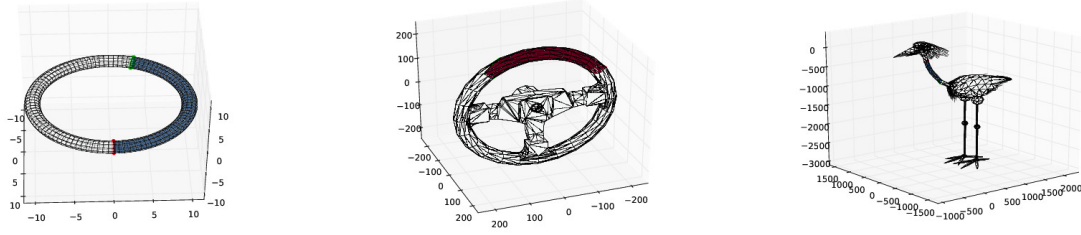


Figure 3: The minimum homology (in terms of the number of triangles) between two cycles is shown shaded in each example above.

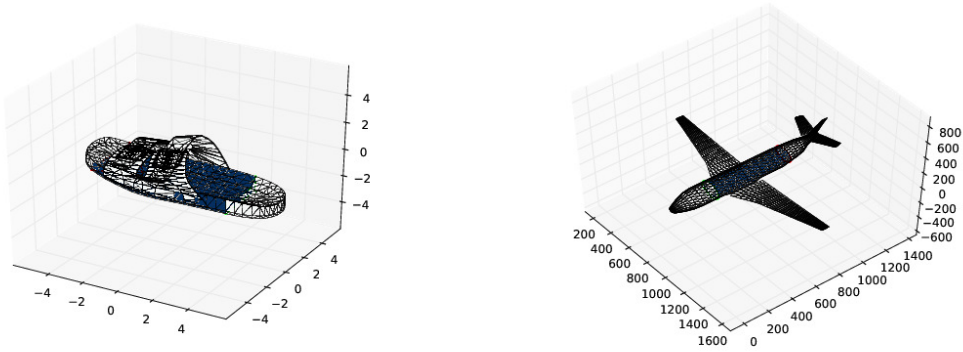


Figure 4: Left: The minimum area homology (shown shaded). Note that because the mesh does not connect the top and bottom of the sandal, the top portion is not included in the area calculation. Right: Again, note the issues in the mesh quality; since the wings are not actually connected to the body of the airplane, the wings are not included in the calculation, although ideally they should be since they would greatly increase the area between these two cycles.

$Dx = DW^{-1}Wx = DW^{-1}\hat{x} = z$ , so that our new problem is  $\arg \min_{\hat{x}} |\hat{x}|_1$  such that  $DW^{-1}\hat{x} = z$ , and we can then let  $x = W^{-1}\hat{x}$ . This lets us use the scikit package to solve our algorithm.

We also have code that computes  $\arg \min \|\hat{x}\|_2$  subject to  $D\hat{x} = \hat{z}$ , using SciPy for computations [JOP\* ]. All our code and our experiments are made available at <http://bitbucket.org/michiexile/minimumhomology>.

## 5.2 Sample outputs

Our images were primarily pulled from the PLY models curated by John Burkardt [Bur] and the Smithsonian X 3D repository [Smia, Smib].

Our own code was implemented in Python, using SciPy [JOP\* ] for linear algebra and Shapely [GBLT] for polygonal area computations. In general, we were able to run it quickly on meshes containing up to 10 000 triangles, which for the Smithsonian meshes required pre-processing with MeshLab [CNR] to simplify and find candidate cycles on the meshes.

In Figures 3 and 4, minimum homologies are shown which were computed using simply a count of the number of triangles included in the homology. Note that as the relevant regions of these meshes have relatively uniform area bounds on their triangles, this preliminary version of the algorithm works quite well; see Section 4.3 for the relevant proofs of why these bounds hold.

Note that in Figure 4, an interesting artifact of the mesh quality was inadvertently revealed in both examples.

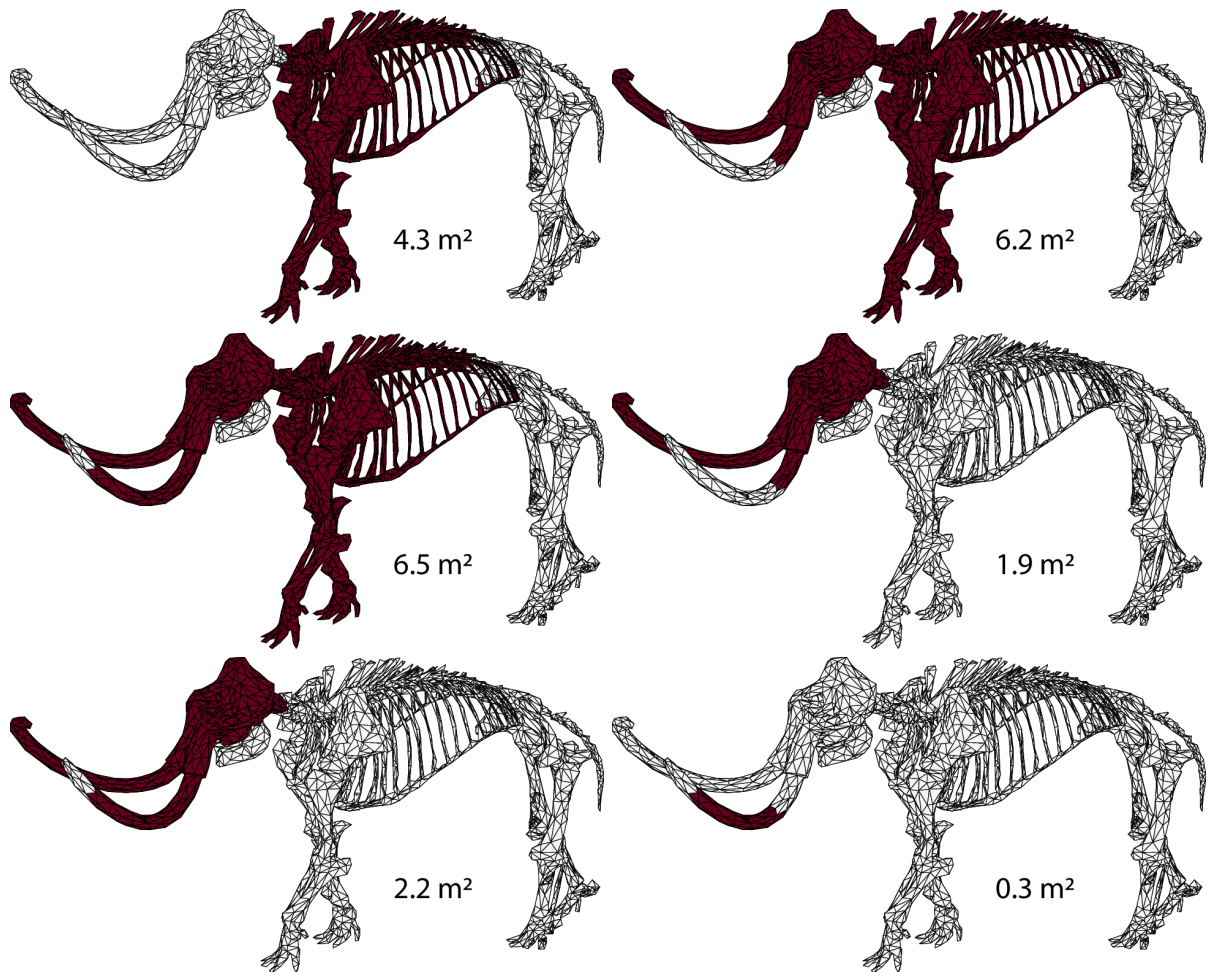


Figure 5: Minimum homology distances between cycles on a Smithsonian scanned model of a Mammoth skeleton

A minimum homology between cycles according to the mesh is shown; however, in each of these examples, the mesh itself consists of disconnected pieces. The airplane actually has the wings detached in the mesh representations, and the sandal's top is a separate piece from the sole. Therefore, our minimum area homology discounts these pieces.

In the Smithsonian meshes, results are far better. One favorable feature is that the meshes are curated and post-processed for 3d-printing, and therefore avoid the kind of self-intersection singularities we see in Figure 4. Another feature is that these meshes are given in a coordinate system with millimeter units, so area computations reflect actual areas on the meshed physical artifacts.

In Figure 5, we show several different minimum area homologies between pairs of input cycles; the area swept by the homology is shown shaded, along with the distances between them given in meters squared. We picked two cycles around the left tusk, and two cycles around the spine, and compute all pairwise distances between these four cycles.

In Figure 6, we can see even more clearly the advantages of using homology, since the indicated cycles are all homologous but not homotopic. We compute all pairwise distances for three different cycles: two single closed loops and a homologous pair of closed loops, mimicking the structure in Figure 1. Again, areas between them are shown shaded, and the distances here are given in thousands of millimeters squared.

Finally, in Figure 7, we demonstrate more minimum homologies between cycles (or collections of cycles, in the case of the two cycles at the base of the claws). We note in this image the results when cycles intersect, such as in the bottom left; this demonstrates a more concrete example of the simple curves shown in Figure 3.

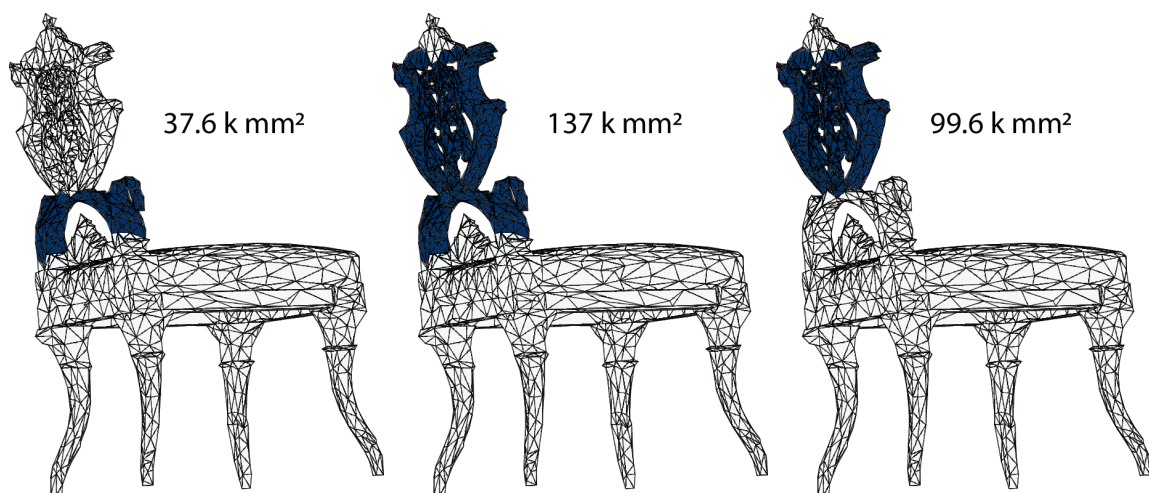


Figure 6: Minimum homology distances between cycles on a Smithsonian scanned model of a renaissance chair. Notice how these bounding chains exemplify the absence of connecting homotopies described in Figure 1.

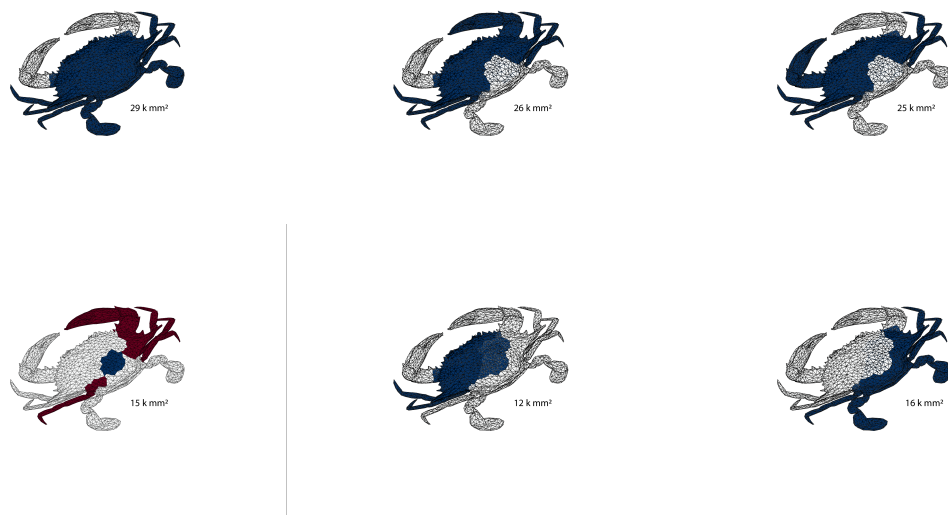


Figure 7: Minimum homology distances between cycles on a Smithsonian scanned model of a crab. In the top row, cycles around each of the claw bases are connected to each other, and to a cycle around the shell. In the bottom row, another cycle around the shell is introduced and connected to each of the other cycles. Notice in particular how the intersecting cycles are handled at the bottom left: this is a case where the cycles do not form a simple cut graph, and the bounding chain is more involved than for our simpler examples.



## References

- [AG95] ALT H., GODAU M.: Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry and its Applications* 5 (1995), 75–91.
- [AKPW14] AHMED M., KARAGIORGOU S., PFOSE D., WENK C.: A comparison and evaluation of map construction algorithms, 2014. Arxiv.org pre-print.
- [Alt09] ALT H.: The computational geometry of comparing shapes. In *Efficient Algorithms*, vol. 5760 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2009, pp. 235–248.
- [AS71] ATIYAH M., SINGER I.: The index of elliptic operators on compact. *Bull. Amer. Math. Soc.* 93 (1971), 119–138.
- [BCP97] BOSMA W., CANNON J., PLAYOUST C.: The magma algebra system i: The user language. *Journal of Symbolic Computation* 24, 3 (1997), 235–265.
- [BKR13] BAUER U., KERBER M., REININGHAUS J.: Clear and compress: Computing persistent homology in chunks. *arXiv preprint arXiv:1303.0477* (2013).
- [Bur] BURKARDT J.: PLY files. URL: <http://people.sc.fsu.edu/~jburkardt/data/ply/ply.html>.
- [CCE\*10] CHAMBERS E. W., COLIN DE VERDIÈRE É., ERICKSON J., LAZARD S., LAZARUS F., THITE S.: Homotopic fréchet distance between curves or, walking your dog in the woods in polynomial time. *Computational Geometry* 43, 3 (2010), 295 – 311. Special Issue on 24th Annual Symposium on Computational Geometry (SoCG’08). URL: <http://www.sciencedirect.com/science/article/pii/S0925772109000637>, doi:10.1016/j.comgeo.2009.02.008.
- [CNR] CNR V. C. L. I.: Meshlab. <http://meshlab.sourceforge.net/>.
- [CW10] COOK A. F., WENK C.: Geodesic Fréchet distance inside a simple polygon. *ACM Transactions on Algorithms* 7 (2010).
- [CW13] CHAMBERS E. W., WANG Y.: Measuring similarity between curves on 2-manifolds via homotopy area. In *Proceedings of the Twenty-ninth Annual Symposium on Computational Geometry* (New York, NY, USA, 2013), SoCG ’13, ACM, pp. 425–434. URL: <http://doi.acm.org/10.1145/2462356.2462375>, doi:10.1145/2462356.2462375.
- [DHK11] DEY T., HIRANI A., KRISHNAMOORTHY B.: Optimal homologous cycles, total unimodularity, and linear programming. *SIAM Journal on Computing* 40, 4 (2011), 1026–1044. URL: <http://epubs.siam.org/doi/abs/10.1137/100800245>, arXiv:<http://epubs.siam.org/doi/pdf/10.1137/100800245>, doi:10.1137/100800245.
- [Don06] DONOHO D. L.: For most large underdetermined systems of linear equations the minimal 1-norm solution is also the sparsest solution. *Communications on pure and applied mathematics* 59, 6 (2006), 797–829.
- [EG08] ELLIS G., GALWAY N.: Homological algebra programming. *Computational group theory and the theory of groups* 470 (2008), 63–74.
- [GAP14] THE GAP GROUP: *GAP – Groups, Algorithms, and Programming, Version 4.7.4*, 2014. URL: <http://www.gap-system.org>.
- [GBLT] GILLIES S., BIERBAUM A., LAUTAPORTTI K., TONNHOFER O.: Shapely. URL: <http://toblerity.org/shapely>.
- [Hat02] HATCHER A.: *Algebraic Topology*. Cambridge University Press, 2002. URL: <http://www.math.cornell.edu/~hatcher/AT/ATpage.html>.
- [HSB66] HIRZEBRUCH F., SCHWARZENBERGER R. L., BOREL A.: *Topological methods in algebraic geometry*, vol. 232. Springer, 1966.
- [JOP\* ] JONES E., OLIPHANT T., PETERSON P., ET AL.: SciPy: Open source scientific tools for Python, 2001–. URL: <http://www.scipy.org/>.
- [KMM04] KACZYNSKI T., MISCHAIKOW K., MROZEK M.: *Computational homology*, vol. 157. Springer, 2004.
- [Law80] LAWSON H.: *Lectures on minimal submanifolds*, vol. 1 of *Mathematics lecture series*. Publish or Perish, 1980. URL: <http://books.google.com/books?id=vlbvAAAAAAAJ>.
- [Mil07] MILNOR J.: A survey of cobordism theory. *Collected Papers of John Milnor: Differential topology* 3 (2007), 291.

- [Mor12] MOROZOV D.: Dionysus, 2012. URL: <http://www.mrzv.org/software/dionysus>.
- [Mun00] MUNKRES J. R.: *Topology*, 2nd ed. Prentice-Hall, 2000.
- [MZ93] MALLAT S. G., ZHANG Z.: Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on* 41, 12 (1993), 3397–3415.
- [Nan12] NANDA V.: Perseus: the persistent homology software, 2012.
- [PRK93] PATI Y., REZAIIFAR R., KRISHNAPRASAD P. S.: Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on* (Nov 1993), pp. 40–44 vol.1. doi:10.1109/ACSSC.1993.342465.
- [PVG\*11] PEDREGOSA F., VAROQUAUX G., GRAMFORT A., MICHEL V., THIRION B., GRISEL O., BLONDEL M., PRETTENHOFER P., WEISS R., DUBOURG V., VANDERPLAS J., PASSOS A., COUNAPEAU D., BRUCHER M., PERROT M., DUCHESNAY E.: Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.
- [RS02] RUBIO J., SERGERAERT F.: Constructive algebraic topology. *Bulletin des Sciences Mathématiques* 126, 5 (2002), 389–412.
- [RZE08] RUBINSTEIN R., ZIBULEVSKY M., ELAD M.: Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit. *CS Technion* (2008), 40.
- [SHI96] SHUM H.-Y., HEBERT M., IKEUCHI K.: On 3d shape similarity. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on* (Jun 1996), pp. 526–531. doi:10.1109/CVPR.1996.517122.
- [SJ05] STEIN W., JOYNER D.: Sage: System for algebra and geometry experimentation. *Communications in Computer Algebra (SIGSAM Bulletin)(July 2005)* (2005). URL: <http://sage.sourceforge.net>.
- [Smia] SMITHSONIAN X 3D: Pergolesi chair, cooper-hewitt, national design museum. URL: <https://3d.si.edu/downloads/64>.
- [Smib] SMITHSONIAN X 3D: Woolly mammoth, national museum of natural history, usnm 23792. URL: <https://3d.si.edu/downloads/55>.
- [TV04] TANGELDER J., VELTKAMP R.: A survey of content based 3d shape retrieval methods. In *Shape Modeling Applications, 2004. Proceedings* (June 2004), pp. 145–156. doi:10.1109/SMI.2004.1314502.
- [TVJA12] TAUSZ A., VEJDEMO-JOHANSSON M., ADAMS H.: javaplex: a research platform for persistent homology. *Book of Abstracts Minisymposium on Publicly Available Geometric/Topological Software* (2012), 7.
- [VJ12] VEJDEMO-JOHANSSON M.: Gap persistence—a computational topology package for gap. *Book of Abstracts Minisymposium on Publicly Available Geometric/Topological Software* (2012), 43.
- [Whi84] WHITE B.: Mappings that minimize area in their homotopy classes. *Journal of Differential Geometry* 20, 2 (1984), 433–446. URL: <http://projecteuclid.org/euclid.jdg/1214439286>.